



GPU ARCHITECTURES AND PROGRAMMING

PROF. SOUMYAJIT DEY

Department of Computer Science and Engineering
IIT Kharagpur

TYPE OF COURSE : New | Elective | UG/PG

COURSE DURATION : 12 weeks (27 Jan' 20 - 17 Apr' 20)

EXAM DATE : 26 Apr 2020

PRE-REQUISITES : Programming and Data Structure, Digital Logic, Computer architecture

INTENDED AUDIENCE : Computer Science, Electronics, Electrical Engg students

INDUSTRIES APPLICABLE TO : NVIDIA, AMD, Google, Amazon and most big-data companies

COURSE OUTLINE :

The course covers basics of conventional CPU architectures, their extensions for single instruction multiple data processing (SIMD) and finally the generalization of this concept in the form of single instruction multiple thread processing (SIMT) as is done in modern GPUs. We cover GPU architecture basics in terms of functional units and then dive into the popular CUDA programming model commonly used for GPU programming. In this context, architecture specific details like memory access coalescing, shared memory usage, GPU thread scheduling etc which primarily effect program performance are also covered in detail. We next switch to a different SIMD programming language called OpenCL which can be used for programming both CPUs and GPUs in a generic manner. Throughout the course we provide different architecture-aware optimization techniques relevant to both CUDA and OpenCL. Finally, we provide the students with detail application development examples in two well-known GPU computing scenarios.

ABOUT INSTRUCTOR :

Prof. Dey joined the Dept. of CSE, IIT Kgp in May 2013. He worked at IIT Patna as Assistant Professor in CSE Dept. from 2012 to 2013. He received a B.E. degree in Electronics and Telecommunication Engg. from Jadavpur University, Kolkata in 2004. and an M.S. followed by PhD degree in Computer Science from Indian Institute of Technology, Kharagpur in 2007 and 2011 respectively. His research interests include 1) Synthesis and Verification of Safe, Secure and Intelligent Cyber Physical Systems, 2) Runtime Systems for Heterogeneous Platforms.

COURSE PLAN :

Week 1: Review of Traditional Computer Architecture – Basic five stage RISC Pipeline, Cache Memory, Register File, SIMD instructions

Week 2: GPU architectures - Streaming Multi Processors, Cache Hierarchy, The Graphics Pipeline

Week 3: Introduction to CUDA programming

Week 4: Multi-dimensional mapping of dataspace, Synchronization

Week 5: Warp Scheduling, Divergence

Week 6: Memory Access Coalescing

Week 7: Optimization examples : optimizing Reduction Kernels

Week 8: Optimization examples : Kernel Fusion, Thread and Block Coarsening

Week 9: OpenCL basics

Week 10: CPU GPU Program Partitioning

Week 11: Application Design : Efficient Neural Network Training/Inferencing

Week 12: Application Design : Efficient Neural Network Training/Inferencing,cont'd